Parameter Modeling of a Two Cross-flow Turbine Array From Experimental Data

Isabel Scherl, Steven L. Brunton, and Brian L. Polagye

***Abstract*—Cross-flow turbines, also known as vertical-axis turbines, use blades that rotate about an axis perpendicular to the incoming flow to convert the kinetic energy in moving fluid to mechanical energy. In this work, the performance of a two-turbine array in a recirculating water channel is modeled using Gaussian process regression. In prior experiments, we optimized "coordinated control" set points (equal tip-speed ratios with an azimuthal phase offset between turbines) to maximize the power output 64 unique geometric configurations with the turbines counter-rotating. While this approach identified promising configurations where turbine pair out-performed geometrically-identical turbines in isolation, the experiments were time-consuming to conduct. In this work, a Gaussian process regression model is initialized with a subset of random points from the geometric configuration space and returns confidence intervals for the full parameter space. Subsequently tested points are then chosen based on the regions of the parameter space model with the highest uncertainty. This is repeated until the model converges (i.e., the model is unchanged with additional points tested). Results are benchmarked against the experimental "truth", but, in future work, would actively guide experimental exploration of high-dimensional spaces.**

***Index Terms*—Cross-flow turbines, vertical-axis turbines, design of experiments**

I. Introduction

CROSS-flow turbines (i.e., vertical-axis turbines in wind) can harness energy from wind and marine currents [1], [2] that have seen a resurgence of interest [3]–[6]. A principal advantage of cross-flow turbines is that dense arrays can outperform equivalent turbines in isolation [7], [8]. This complements other benefits of dense arrays, including increased power output per area [9].

Exploring dynamics in the arrays presents a unique challenge because the dimensionality of the parameter space increases exponentially with the number of turbines. Given this, traditional uniform sampling across parameters quickly becomes intractable and it becomes necessary to explore alternative methods. Optimization schemes have been used to explore similarly large parameter spaces [10] and, while an excellent solution to find optimality, they cannot provide full information about how various parameters affect performance. Knowledge about the impact of differenet variables on underlying dynamics can be as valuable as the optimal solution for array design. For example, this information could use this to identify designs where performance will not significantly vary if control or environmental parameters are perturbed. Here we show that it is possible to efficiently characterize an array performance parameter space using Gaussian process regression, a method that has recently gained popularity to solve similar problems. Gaussian process regression has been used to explore system dynamics in a variety of fields. These include, vortex-induced vibrations [11], phytoplankton sampling in the ocean [12], spectroscopy [13], learning partial differential equations [14].

In this work, we show how Gaussian process regression can be used to predict the relative performance of an array using the performance of arrays with similar but different configurations.

II. Methods

A. Experiments

The experimental data used in this work is described in detail in [15]. Experiments were conducted in a recirculating flume at the Bamfield Marine Science Centre. The experimental set-up consisted of two cross-flow turbines: a stationary turbine instrumented with force sensors to measure thrust and torque and a mobile turbine instrumented with a torque sensor. Both turbines has their rotational motion regulated by servomotors to ensure constant rotational velocity. A photo of the two turbines operating is shown in Figure 1(b). The mobile turbine was mounted to a robotic gantry system to move it into a variety of geometric positions. The geometric positions tested are shown in Figure 1(a).

B. Performance Metrics

The coefficient of performance for an individual turbine is the ratio of the power produced to the kinetic power in the free-stream passing through the turbine's projected area and expressed as

$$C_p = \frac{P}{\frac{1}{2}\rho U_\infty^3 HD} = \frac{\omega\tau}{\frac{1}{2}\rho U_\infty^3 HD}, \tag{1}$$

where P is turbine's mechanical power, ρ is the fluid density, U_∞ is the freestream flow velocity, H is the turbine height, D is the turbine diameter, ω is the turbine rotation rate, and τ is the turbine torque. We evaluated array performance relative to the sum of the maximum performances of the two turbines under similar environmental conditions but in isolation as

$$\kappa = \frac{C_{P,1} + C_{P,2}}{C^*_{P,1} + C^*_{P,2}} \tag{2}$$

where κ is the interaction factor ($\kappa < 1$ for detrimental interactions and $\kappa > 1$ for beneficial interactions) and a star(*) denotes performance in isolation. We note that due to minor differences in support structures, measurement, and control hardware, $C^*_{P,1} \neq C^*_{P,2}$.

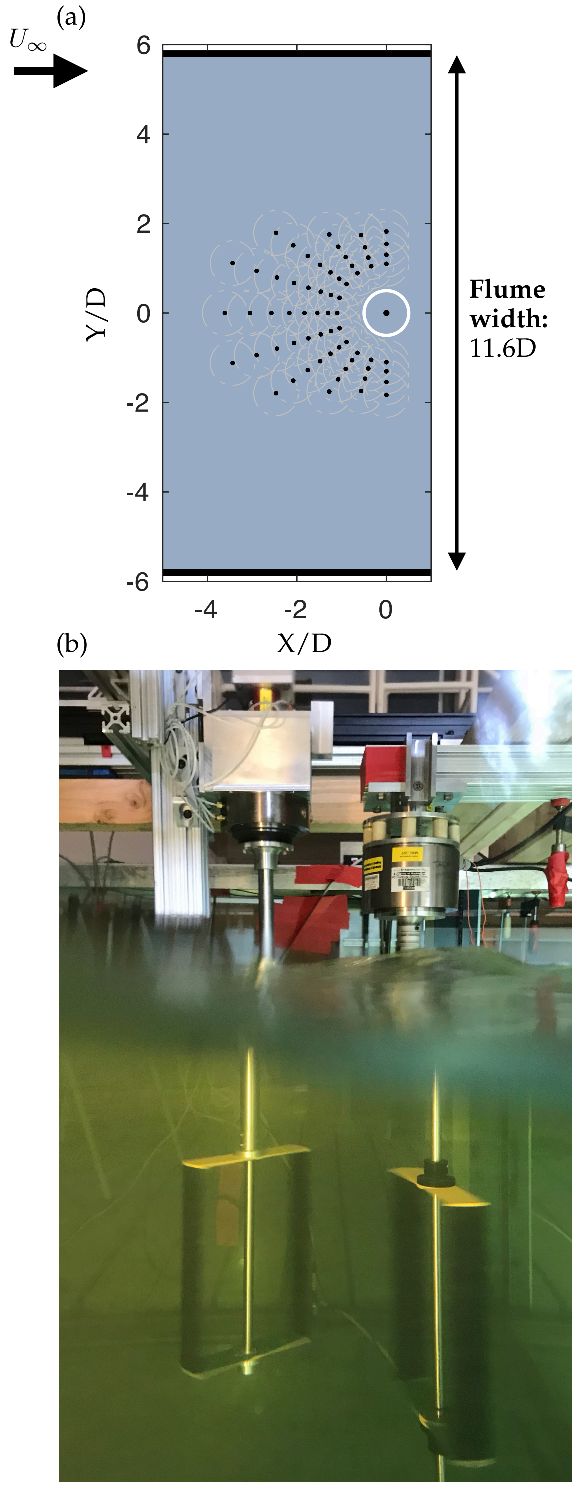

Fig. 1. (a) Each array configuration tested, where one turbine was fixed at $X/D = Y/D = 0$ and the other turbine was tested at each prescribed polar grid location (b) photo of both turbines operating in the Bamfield Marine Science Centre flume

The turbine arrays were operated under "coordinated control" where the angular velocities, or tip-speed ratios given by

$$\lambda = \frac{\omega R}{U_\infty}, \tag{3}$$

where R is the turbine radius, of the two turbines are locked to the same value ($\lambda = \lambda_1 = \lambda_2$). For counter-rotating turbines, the phase difference is defined as

$$\phi = -\theta_1 - \theta_2, \tag{4}$$

where θ_1 and θ_2 are the angular positions of mobile and fixed turbines respectively and $\theta = 0$ when one blade is pointing directly upstream. This results in a two-parameter space consisting of λ and ϕ. A closed-loop controller is used to maintain a constant ϕ while testing a specific pair of parameter values. The optimized performance values used to train the Gaussian process regression model were acquired using a hardware-in-the-loop optimizer built around a Nelder-Mead simplex [16].

C. *Gaussian Process Regression*

A Gaussian process is a stochastic process where each dimension can be described as a Gaussian or normal distribution. Gaussian process regression (GPR) uses a collection of Gaussian processes to describe a stochastic process [17]. This method is nonparametric which means it does not require knowledge of the system. Gaussian process regression generates a model the parameter space and quantifies the uncertainty in its predictions of the parameter space. This quantification can be a useful tool to increase experimental efficiency by choosing the region with the highest uncertainty to sample next.

The model in this work was implemented using the MATLAB Statistics and Machine Learning Toolbox. We specified a squared exponential kernel function and a linear basis function. The process we use to train and evaluate the Gaussian process regression model is as follows. We initialize our model with the known interaction factor at $n = 5$ randomly selected training locations. In other words, as if we had identified optimal coordinated control strategies at 5 of the 64 possible array layouts. The model then produces an estimate of the interaction factor at the remaining locations and the location with the highest uncertainty is then added to the training data (experimentally, this would be the next point tested). The model is then re-evaluated to determine the next point to add to the training data. This process continues until all points have been added to the training data.

To generalize the statistical efficacy of the model as a function of the number of points in the training data, the modeling procedure is repeated m times with different initial training data. The mean relative error for the GPR model is defined as

$$\left\langle \frac{||\bar{\kappa} - \bar{\kappa}_{predicted}||_2}{||\bar{\kappa}||_2} \right\rangle_m \tag{5}$$

where we are comparing the true interaction factors ($\bar{\kappa}$) to the predicted interaction factor ($\bar{\kappa}_{predicted}$) averaged over m random initializations and the bar denotes that we are looking at a vector of interaction factors over the parameter space. To determine statistical convergence, we evaluated Equation 5 for $m = 10, 30, 100, 300$ initializations.

III. Results and Discussion

The interaction factor predictions can be seen in Figure 2 for one set of randomly generated initial

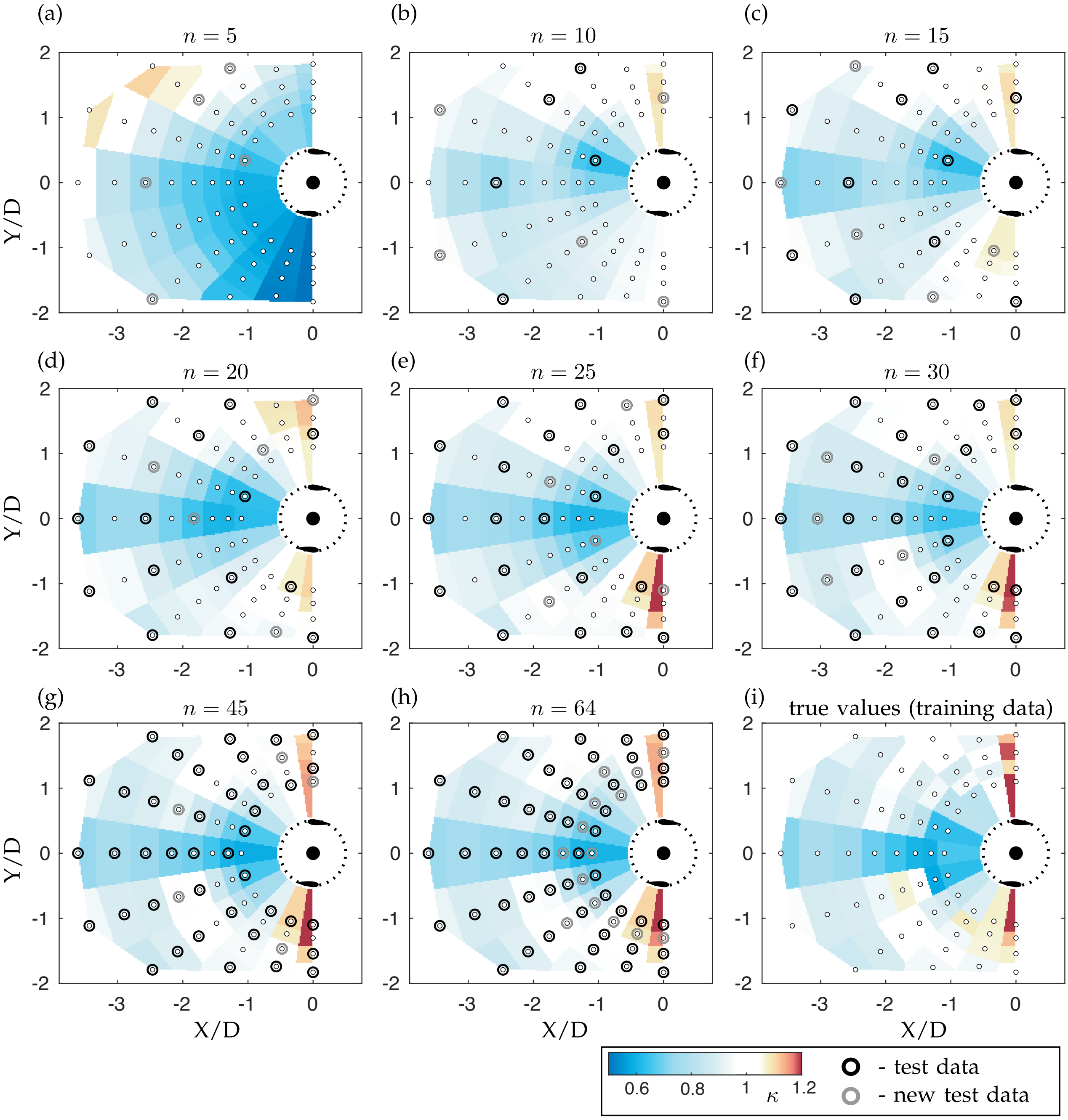

Fig. 2. (a)-(h) Interaction factor prediction for GPR models from circled points with $n = 5, 10, 15, 20, 25, 45, 64$ test points, respectively, where circled points are used to train the model and the grey points have been newly added since the previously shown case (i) training data for each location

training data. The fidelity of the interaction factor prediction improves as we increase the number of points in the training data. The most important aspects of the parameter space emerge when the model uses fewer than one third of the total points at $n \approx 20$ (Fig. 2(c)). As the number of training points is further increased, the model more closely matches the true performance, with the models at $n = 45$ and $n = 64$ nearly indistinguishable from the true parameter space. Since we are able to accurately predict interaction factors using neighboring points, it is likely that we over-sampled when originally acquiring the experimental data, particularly when the turbines are in similar streamwise positions and in close proximity. Consequently, due to their low uncertainty, these are the final points added to the training data in the example in Figure 2.

The statistical performance under m realizations are shown in Figure 3. The results from the smaller values of $m = 10, 30$ are not converged and "jagged", whereas there is little deviation between the mean relative error at $m = 100$ initializations and $m = 300$ initializations. It should be noted that even with small m values, the results do not vary widely. The inter-quartile range (IQR) for each value of n at $m = 100$ is shown in Figure 3(b). In this plot, the IQR is larger with fewer points due to increased variance in which points can be selected (i.e., if 5 out 64 points are used as training data then there are more combinations of points available than when forty points are used as training data). The IQRs for a random progression (i.e., n points chosen entirely at random) and GPR progression overlap until approximately $n \approx 30$ where the GPR consistently

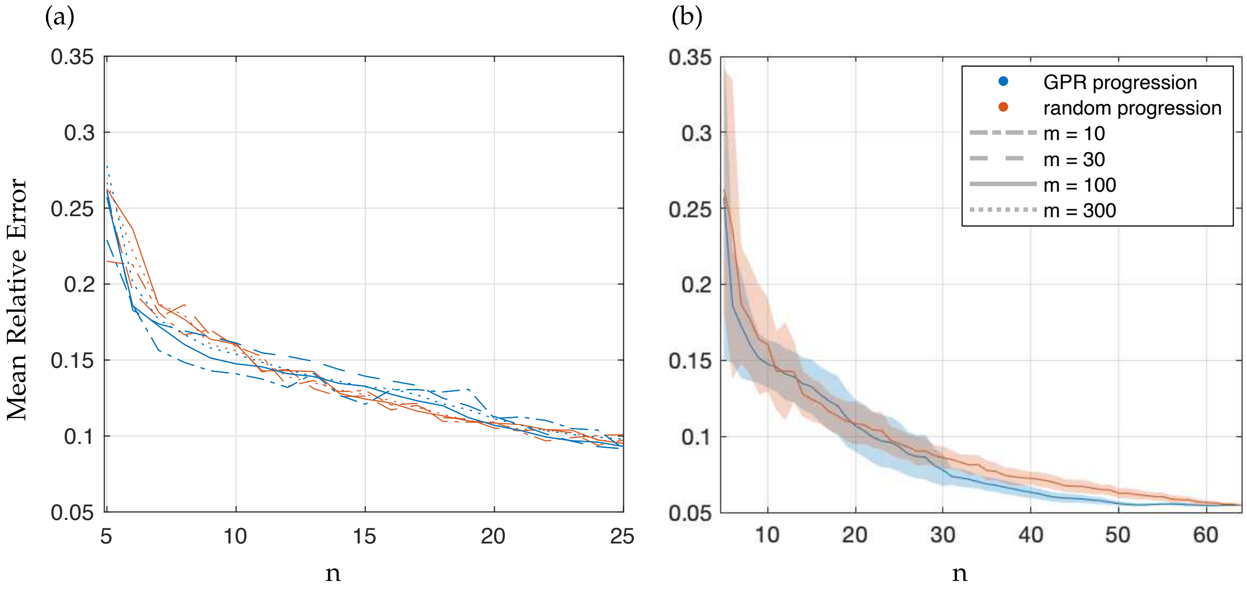

Fig. 3. (a) Mean relative error of Gaussian process regression (GPR) progressed models (blue) and randomly progressed models (red) for various iterations of random initializations tested ($m = 10, 30, 200, 300$) (b) Mean relative error and inter-quartile range (IQR) for $m = 100$

outperforms random progression. In summary, choosing subsequent points based on the uncertainty in the model is more efficient than random sampling with an equal number of points and *significantly* more efficient than uniform sampling.

IV. Conclusion & Future Work

In this work we show how Gaussian process regression modeling can be used to efficiently predict the performance of neighboring geometries in arrays of cross-flow turbines. This is an exciting result that shows the promise of using this design of experiments approach to model the high dimensional space of larger turbine arrays. Future work will be to implement this approach in real-time as a model-based, hardware-in-the-loop approach to data collection.

References

[1] S. Salter, “Are nearly all tidal stream turbine designs wrong?” in *4th International Conference on Ocean Energy*, 2012, pp. 1–7.

[2] S. Eriksson, H. Bernhoff, and M. Leijon, “Evaluation of different turbine concepts for wind power,” *renewable and sustainable energy reviews*, vol. 12, no. 5, pp. 1419–1434, 2008.

[3] C. M. Parker and M. C. Leftwich, “The effect of tip speed ratio on a vertical axis wind turbine at high reynolds numbers,” *Experiments in Fluids*, vol. 57, no. 5, p. 74, 2016.

[4] R. Dunne and B. J. McKeon, “Dynamic stall on a pitching and surging airfoil,” *Experiments in Fluids*, vol. 56, no. 8, p. 157, 2015.

[5] P. Bachant and M. Wosnik, “Characterising the near-wake of a cross-flow turbine,” *Journal of Turbulence*, vol. 16, no. 4, pp. 392–410, 2015.

[6] P. Bachant, M. Wosnik, B. Gunawan, and V. S. Neary, “Experimental study of a reference model vertical-axis cross-flow turbine,” *PloS one*, vol. 11, no. 9, p. e0163799, 2016.

[7] I. D. Brownstein, M. Kinzel, and J. O. Dabiri, “Performance enhancement of downstream vertical-axis wind turbines,” *Journal of Renewable and Sustainable Energy*, vol. 8, no. 5, p. 053306, 2016.

[8] B. Strom, S. L. Brunton, and B. Polagye, “Advanced control methods for cross-flow turbines,” *International Marine Energy Journal*, vol. 1, no. 2 (Nov), pp. 129–138, 2018.

[9] J. O. Dabiri, “Emergent aerodynamics in wind farms,” *Physics today*, vol. 67, no. 10, pp. 66–67, 2014.

[10] B. Strom, I. Scherl, S. Brunton, and B. Polagye, “Geometric and control optimization of an array of two cross-flow turbines,” *Bulletin of the American Physical Society*, vol. 63, 2018.

[11] D. Fan, G. Jodin, T. Consi, L. Bonfiglio, Y. Ma, L. Keyes, G. E. Karniadakis, and M. S. Triantafyllou, “A robotic intelligent towing tank for learning complex fluid-structure dynamics,” *Science Robotics*, vol. 4, no. 36, 2019.

[12] T. O. Fossum, G. M. Fragoso, E. J. Davies, J. E. Ullgren, R. Mendes, G. Johnsen, I. Ellingsen, J. Eidsvik, M. Ludvigsen, and K. Rajan, “Toward adaptive robotic sampling of phytoplankton in the coastal ocean,” *Science Robotics*, vol. 4, no. 27, 2019.

[13] T. Chen, J. Morris, and E. Martin, “Gaussian process regression for multivariate spectroscopic calibration,” *Chemometrics and Intelligent Laboratory Systems*, vol. 87, no. 1, pp. 59–71, 2007.

[14] M. Raissi and G. E. Karniadakis, “Hidden physics models: Machine learning of nonlinear partial differential equations,” *Journal of Computational Physics*, vol. 357, pp. 125–141, 2018.

[15] I. Scherl, B. Strom, S. L. Brunton, and B. L. Polagye, “Geometric and control optimization of a two cross-flow turbine array,” *Journal of Renewable and Sustainable Energy*, vol. 12, no. 6, p. 064501, 2020.

[16] J. A. Nelder and R. Mead, “A simplex method for function minimization,” *The computer journal*, vol. 7, no. 4, pp. 308–313, 1965.

[17] C. E. Rasmussen, “Gaussian processes in machine learning,” in *Summer school on machine learning*. Springer, 2003, pp. 63–71.